

IERG 4210 Tutorial 4 – Phase 2B

Part 2: Web Development

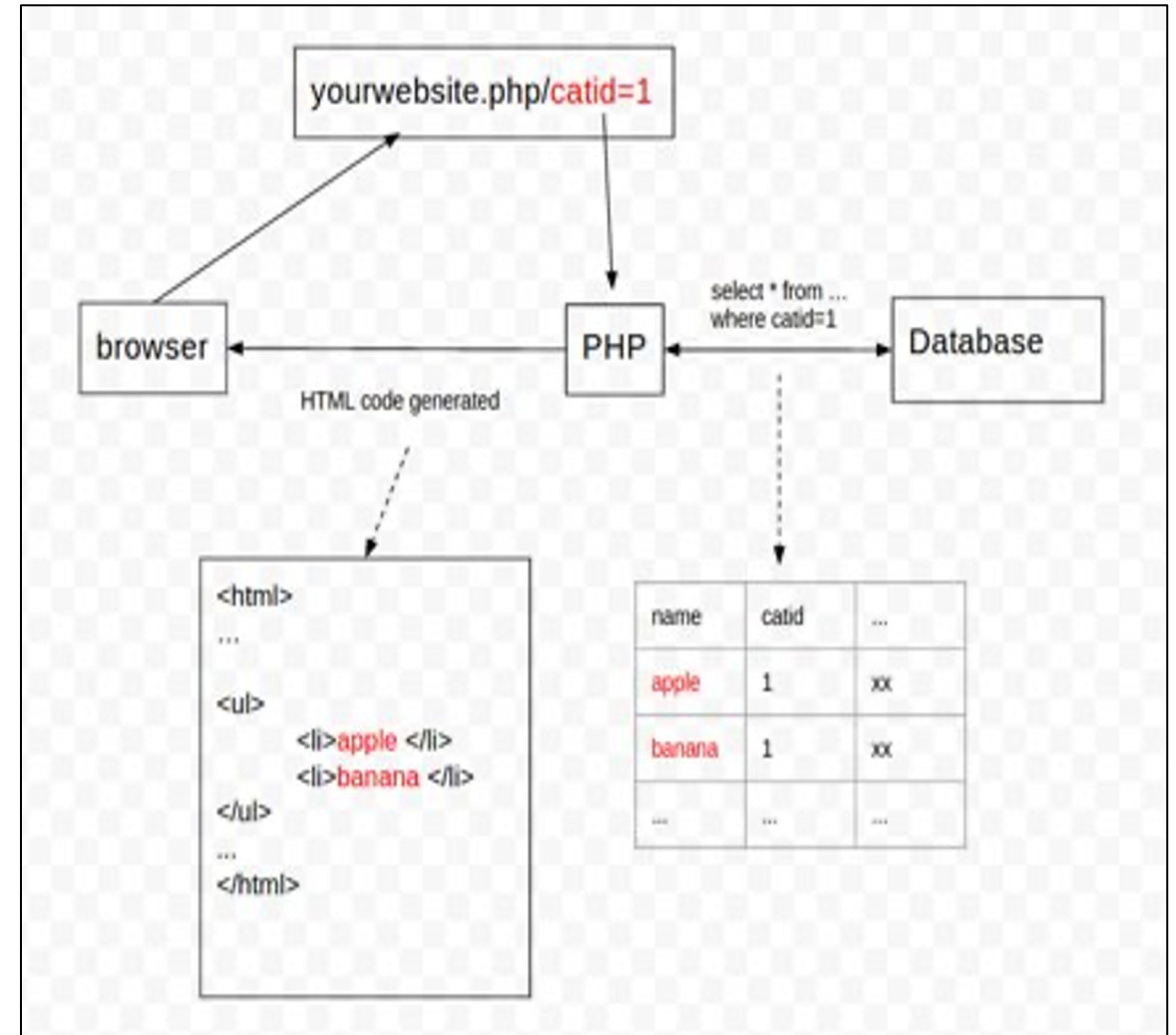
Xiao Yi modified according to previous TA **Siyue Xie**'s slides
yx019@ie.cuhk.edu.hk

Outline

- Web Development & Database Manipulation
- Server-side Request Handling

Web Development & Database Manipulation

- ▶ Workflow
 - ▶ User sends requests
 - ▶ Server handles requests
 - ▶ Query records from database
 - ▶ Or delete/ update/ etc.
 - ▶ Other preprocessing
 - ▶ Render HTML by populating preprocessed info
 - ▶ Display the webpage to the user



Web Development & Database Manipulation

- ▶ Admin Panel
 - ▶ Collect requests of adding /editing category / product info through HTML form
 - ▶ Server processes the requests and manipulates the database

IERG4210 Shop - Admin Panel (Demo)

New Category

Name

Submit

New Product

Category *

Name *

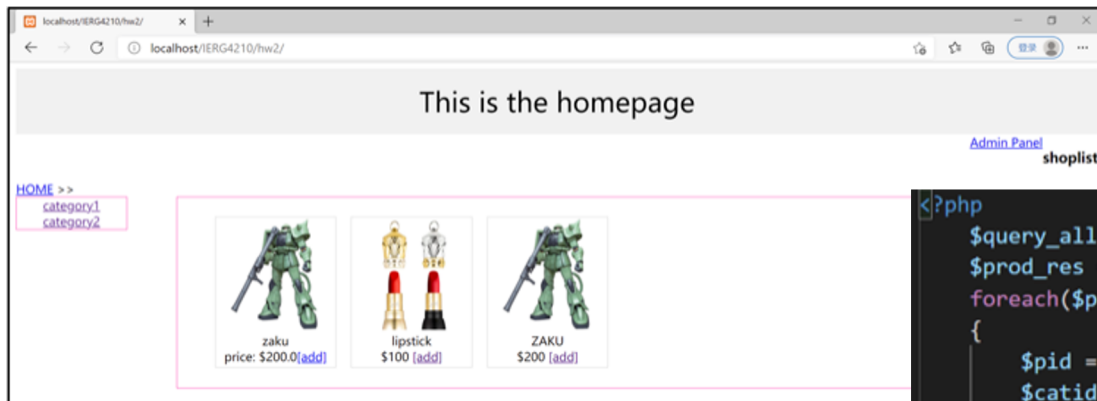
Price *

Description

Image *
Choose File No file chosen
Submit

Web Development & Database Manipulation

- ▶ Homepage
 - ▶ Query records from database
 - ▶ Populate information and render the HTML page



```
<?php
$query_all = $db->query("SELECT pid, catid, name, price FROM products LIMIT 20;");
$prod_res = $query_all->fetchAll();
foreach($prod_res as $prod_elm)
{
    $pid = $prod_elm['pid'];
    $catid = $prod_elm['catid'];
    $prod_name = $prod_elm['name'];
    $prod_price = $prod_elm['price'];
    ?>
    <li>
    <a href='details.php?itn=<?php echo htmlspecialchars($prod_name) ?>&catid=<?php echo htmlspecialchars($catid) ?>'>
    <img src='./img/thumbnail/<?php echo $catid.'/'.'$pid.'.png'; ?>'>
    </a><br>
    <?php echo $prod_name."<br>price: ".$prod_price; ?><a href=#>[add]</a><br>
    </li>
    <?php
}
```

Server-Side Request Handling

- ▶ “GET” request
 - ▶ requests a specific resource from server
 - ▶ parameters are appended in URL
 - ▶ Security issues
 - ▶ response can be cached, bookmarked; remained in browser history; etc.
- ▶ “POST” request
 - ▶ submits data to server
 - ▶ parameters are encoded and invisible in the URL
 - ▶ support file uploading

Server-side Request Handling

► Example: HTML Form

```
<fieldset>
<legend>New Product</legend>
<form id="prod_insert" method="POST" action="admin-process.php?action=xxx"
enctype="multipart/form-data">
    <label for="prod_catid">Category *</label>
    <div><select id="prod_catid" name="catid"></select></div>
    <label for="prod_name">Name *</label>
    <div><input id="prod_name" type="text" name="name" required="true"
pattern="^[\\w\\- ]+$" /></div>

    ...

    <label for="prod_name">Image *</label>
    <div><input type="file" name="file" required="true" accept="image/jpeg" /></div>
    <input type="submit" value="Submit" />
</form>
</fieldset>
```

The image shows a visual rendering of the HTML form code provided. It is titled "New Product" and contains the following elements: a "Category *" dropdown menu, a "Name *" text input field, a "Price *" text input field, a "Description" text area, and an "Image *" section. The "Image *" section includes a "Choose File" button, the text "No file chosen", and a "Submit" button.

Parse Requests on Server-side

- ▶ PHP superglobal array
 - ▶ `$_POST`, `$_GET`, `$_REQUEST`, `$_SERVER`, `$_SESSION`
 - ▶ User inputs will be stored in superglobal arrays
- ▶ Request parameters
 - ▶ `action= "admin-process.php?action=xxx"`
 - ▶ xxx will be stored in the superglobal array `$_GET` and `$_REQUEST`
 - ▶ E.g., `xxx == $_REQUEST["action"]`
- ▶ Manipulate the database on server-side via PHP
 - ▶ PHP PDO

PHP Request Handling

- ▶ Check the type of request
 - ▶ E.g., `if ($_SERVER["REQUEST_METHOD"] === "POST")`
- ▶ Implement interfaces to handle user request
 - ▶ `function ierg4210_prod_insert()`
 - ▶ `function ierg4210_category_edit() ...`
 - ▶ **use `call_user_func` to call certain function according to action parameter**
 - ▶ E.g., `call_user_func("ierg_4210_".$_REQUEST["action"])`

PHP File Uploading Management (1/2)

- ▶ Client-side

- ▶ Add `enctype="multipart/form-data"` in form tag to allow file uploading

- ▶ Server-side

- ▶ `$_FILES` is a superglobal array which stores all information of an uploaded file
 - ▶ `$_FILES["file"]["error"]` – return the error message if any. 0 means no error
 - ▶ `$_FILES["file"]["name"]` – the original file name
 - ▶ `$_FILES["file"]["type"]` – type of the file. e.g., image/jpeg for a JPEG file
 - ▶ `$_FILES["file"]["size"]` – size of the file in bytes
 - ▶ `$_FILES["file"]["tmp_name"]` – file path of the temporarily stored file
 - ▶ check file format, size (format: jpg/gif/png, size <= 10MB)
 - ▶ Modify php.ini to allow file uploading (post_max_size, file_uploads)

PHP File Uploading Management (2/2)

- ▶ Database Manipulation
 - ▶ Insert records/ the uploaded file into the database
- ▶ Post-processing
 - ▶ `$filename = $_FILES["file"]["tmp_name"]`
 - ▶ `move_uploaded_file($filename, $destination)`
 - ▶ Check the permission of storing a file in the destination folder (x+w)
 - ▶ Response of a successful uploading:
 - ▶ `header('Location: admin.html'); exit();`

PHP Input Validation

► Regex in PHP

- `preg_match($pattern, $subject)`

- E.g., price must be a valid number; names don't have special letters...

- Some rules:

- outside square brackets

- `\` general escape character
 - `^` assert start of subject
 - `$` assert end of subject
 - `.` match any character except newline
 - `|` start of alternative branch
 - `? * + { } ()`

- inside square brackets

- `\` general escape character
 - `^` negate the class, but only if the first character
 - `-` indicates character range

```
if (!preg_match('/^\d*$/ ', $_POST['catid']))  
    throw new Exception("invalid-catid");
```

Update Your Website

- ▶ Rewrite your website using PHP
 - ▶ Parse requests
 - ▶ Update your webpage based on database records
 - ▶ Update database according to user requests/ submissions/ etc.
- ▶ Other tricks
 - ▶ Use template engine to render HTML pages
 - ▶ a good separation between application logic and display logic
 - ▶ back-end developers and front-end developers can collaborate on the same areas of the website without having to step around each other's code
 - ▶ Options: <http://acmeextension.com/best-templating-engine/>